

contents

PREFACE xiii

ACKNOWLEDGEMENTS xvi

NOTE TO THE STUDENT xviii

1. BACKGROUND 1

1. Machine Structure 2

2. Evolution of the Components of a Programming System 4

1. Assemblers 4

2. Loaders 4

3. Macros 6

4. Compilers 7

5. Formal Systems 7

3. Evolution of Operating Systems 8

4. Operating System User Viewpoint: Functions 10

5. Operating System User Viewpoint: Batch Control Language 11

6. Operating System User Viewpoint: Facilities 14

7. Summary 14

Questions 16

2. MACHINE STRUCTURE, MACHINE LANGUAGE, AND ASSEMBLY LANGUAGE 21

1. General Machine Structure 21

1. General Approach to a New Machine 23

2. Machine Structure—360 and 370 25

1. Memory 25 2. Registers 25 3. Data 27

4. Instructions 29 5. Special Features 35

2. Machine Language 35

1. Long Way, No Looping 36

2. Address Modification Using Instructions as Data 38

3. Address Modification Using Index Registers	40
4. Looping	41
3. Assembly Language	43
1. An Assembly Language Program	43
2. Example Using Literals	45
4. Summary	47
Questions	48
3. ASSEMBLERS	59
1. General Design Procedure	60
2. Design of Assembler	60
1. Statement of Problem	60
2. Data Structure	62
3. Format of Data Bases	65
4. Algorithm	73
5. Look for Modularity	77
3. Table Processing: Searching and Sorting	80
1. Linear Search	81
2. Binary Search	82
3. Sorting	84
1. Interchange Sort	84
2. Shell Sort	86
3. Bucket Sort	86
4. Radix Exchange Sort	88
5. Address Calculation Sort	88
6. Comparison of Sorts	90
4. Hash or Random Entry Searching	91
4. Summary	95
5. Epilog	95
Questions	98
4. MACRO LANGUAGE AND THE MACRO PROCESSOR	111
1. Macro Instructions	112
2. Features of a Macro Facility	114
1. Macro Instruction Arguments	114
2. Conditional Macro Expansion	117
3. Macro Calls within Macros	119
4. Macro Instructions Defining Macros	121
3. Implementation	122
1. Implementation of a Restricted Facility: A Two-Pass Algorithm	123
2. A Single-Pass Algorithm	127
3. Implementation of Macro Calls within Macros	133
4. Implementation within an Assembler	136
4. Summary	142
Questions	143

5. LOADERS	149
1. Loader Schemes	150
1. "Compile-and-Go" Loaders	150
2. General Loader Scheme	151
3. Absolute Loaders	152
4. Subroutine Linkages	154
5. Relocating Loaders	156
6. Direct-linking Loaders	160
7. Other Loader Schemes—Binders, Linking Loaders, Overlays, Dynamic Binders	164
2. Design of an Absolute Loader	167
3. Design of a Direct-Linking Loader	168
1. Specification of Problem	169
2. Specification of Data Structures	176
3. Format of Data Bases	178
4. Algorithm	181
4. Summary	186
Questions	188
 6. PROGRAMMING LANGUAGES	 199
1. Importance of High Level Languages	200
2. Features of a High Level Language	201
3. Data Types and Data Structures	202
1. Character String	203
2. Bit String—Boolean	204
3. Data Structures	205
4. Storage Allocation and Scope of Names	206
1. Storage Classes	206
2. Block Structure	206
5. Accessing Flexibility	207
1. Pointers	208
2. Label Variables and Label Arrays	208
6. Functional Modularity	210
1. Procedures	211
2. Recursion	212
7. Asynchronous Operation	212
1. Conditions	212
2. Signals	213
3. Multitasking	215
8. Extensibility and Compile-Time Macros	215
9. Miscellaneous	216
10. Summary	216
Questions	217

7. FORMAL SYSTEMS AND PROGRAMMING LANGUAGES: AN INTRODUCTION 227

- 1. Uses of Formal Systems in Programming Languages 227**
 - 1. Language Specification 228**
 - 2. Syntax-directed Compilers 228**
 - 3. Complexity Structure Studies 229**
 - 4. Structure Analysis 229**
- 2. Formal Specification 230**
 - 1. Approaching a Formalism 230**
 - 2. Development of Formal Specification 231**
- 3. Formal Grammars 233**
 - 1. Examples of Formal Grammars 235**
 - 2. The Derivation of Sentences 236**
 - 3. Sentential Forms and Sentences 236**
- 4. Hierarchy of Languages 237**
- 5. Backus-Naur Form—Backus Normal Form—BNF 239**
- 6. Canonic Systems 240**
 - 1. Example: Syntax Specification 244**
 - 2. Specification of Translation 246**
 - 3. Recognition and Translation Algorithm 247**
- 7. Canonic Systems and Formal Systems 251**
- 8. Summary 256**
- Questions 258**

8. COMPILERS 265

PART 1

- 1. Statement of Problem 265**
 - 1. Problem No. 1—Recognizing Basic Elements 266**
 - 2. Problem No. 2—Recognizing Syntactic Units and Interpreting Meaning 268**
 - 3. Intermediate Form 269**
 - 1. Arithmetic Statements 269**
 - 2. Nonarithmetic Statements 270**
 - 3. Nonexecutable Statements 271**
 - 4. Problem No. 3—Storage Allocation 271**
 - 5. Problem No. 4—Code Generation 273**
 - 1. Optimization (Machine-independent) 275**
 - 2. Optimization (Machine-dependent) 275**
 - 3. Assembly Phase 277**
 - 6. General Model of Compiler 277**

PART 2

- 2. Phases of the Compiler 279**
 - 1. Lexical Phase 279**
 - 1. Tasks 279**
 - 2. Data Bases 279**
 - 3. Algorithm 282**
 - 4. Example 283**

2. Syntax Phase	283		
1. Data Bases	285	2. Algorithm	286
3. Example	286		
3. Interpretation Phase	287		
1. Data Bases	288	2. Algorithm	289
3. Example	290		
4. Optimization	293		
1. Data Bases	293	2. Algorithm	298
5. Storage Assignment	301		
1. Data Bases	301	2. Algorithm	302
3. Example	304		
6. Code Generation	306		
1. Data Bases	306	2. Algorithm	307
7. Assembly Phase	313		
1. Data Bases	313	2. Algorithm	313
8. Passes of a Compiler	314		
9. Preview	314		

PART 3

3. Data Structures	316
1. Statement of Problem	316
2. Implementation	320
4. Recursion, Call, and Return Statements	321
5. Storage Classes—Use	326
1. Static Storage	326
2. Automatic Storage	327
3. Internal Controlled Storage	327
4. External Controlled Storage	328
5. Based Storage	328
6. Implementation	328
1. Static Storage	328
2. Automatic Storage	330
3. Controlled and Based Storage	330
7. Block Structure	331
1. Accessing Information for Block Structure	333
2. Storage Allocation for Block Structure	335
8. Nonlocal Go To's	338
9. Interrupts	339
10. Pointers	339
11. Summary	341
Questions	342

9. OPERATING SYSTEMS 349

PART 1

1. I/O Programming: Multiple Processors and Interrupt Mechanisms	350
1. Evolution of Multiple Processor System	350
2. I/O Programming	352
3. I/O Processor Structure	353

- 4. Examples of I/O Programs 355
- 5. Communications between the CPU and the Channel 357
- 6. Interrupt Structure and Processing 359
- 7. Example of I/O Interrupt Processing 362
- 8. Multiple Processors 365

PART 2

- 2. Memory Management 366
 - 1. Single Contiguous Allocations 367
 - 2. Partitioned Allocation 367
 - 3. Relocatable Partitioned Allocation 370
 - 4. Paged Allocation 373
 - 5. Demand Paged Allocation 376
 - 6. Segmented Allocation 378
 - 7. Segmented-Paged Allocation 386

PART 3

- 3. Processor Management 388
 - 1. Scheduler 389
 - 2. Traffic Controller 392
 - 3. Race Condition 392
 - 4. Stalemates 393
 - 5. Multiprocessor Systems 395

PART 4

- 4. Device Management 401
 - 1. Device Characteristics 401
 - 2. Device Management Techniques 405

PART 5

- 5. Information Management 407
 - 1. Development of File Systems 408
 - 2. Structure of a General File System 409
 - 3. Example of a File System 409
 - 4. Features of a General File System 413
 - 5. General File System Model Revisited 421
 - 6. Segmentation 424
 - 7. MULTICS File System and the General Model 430
- 6. Summary 430
- Questions 433

10. BIBLIOGRAPHY AND SUGGESTIONS FOR FURTHER READING 441

appendix A 360 SPECIFICATION 451

appendix B LINKAGE CONVENTIONS 465

INDEX 471